Smolign: A Spatial Motifs Based Protein Multiple Structural Alignment Method

Hong Sun, Ahmet Sacan, Hakan Ferhatosmanoglu, and Yusu Wang

Abstract—Availability of an effective tool for protein multiple structural alignment (MSTA) is essential for discovery and analysis of biologically significant structural motifs that can help solve functional annotation and drug design problems. Existing MSTA methods collect residue correspondences mostly through pairwise comparison of consecutive fragments, which can lead to suboptimal alignments, especially when the similarity among the proteins is low.

We introduce a novel strategy based on: building a contact-window based motif library from the protein structural data, discovery and extension of common alignment seeds from this library, and optimal superimposition of multiple structures according to these alignment seeds by an enhanced partial order curve comparison method. The ability of our strategy to detect multiple correspondences simultaneously, to catch alignments globally, and to support flexible alignments, endorse a sensitive and robust automated algorithm that can expose similarities among protein structures even under low similarity conditions. Our method yields better alignment results compared to other popular MSTA methods, on several protein structure datasets that span various structural folds and represent different protein similarity levels.

A web-based alignment tool, a downloadable executable, and detailed alignment results for the datasets used here are available at http://sacan.biomed.drexel.edu/Smolign and http://bio.cse.ohio-state.edu/Smolign

Index Terms—Protein structure, multiple structure alignment, partial order curve comparison, structural motif library, secondary structure elements (SSE), distance map, contact map, HOMSTRAD.

1 Introduction

Proteins carry out their specific biological roles through interaction with other proteins or other macro-molecules. This interaction is determined largely by the three dimensional structures of molecules. Therefore, an important direction toward understanding how proteins function is to study and analyze their structures. In particular, since many structurally similar proteins have a common evolutionary origin, one fundamental task involved in such an analysis is the structural alignment problem, where the proteins are superimposed in order to find the similarities and differences in their structures. Alignment and comparison of protein structures can help discover biologically significant structural motifs and reveal distant evolutionary relationships that may not be detectable from the sequence information alone.

In recognition of the important relationship between structure and function, there has been a large volume of research on the structural alignment problem over the past twenty years. Early research focused primarily on the *pairwise structural alignment problem* [1], where an optimal superposition of two protein structures is sought such as to minimize a given geometric distance measure. The quality of an alignment is generally quantified by two parameters: the number of corresponding residues among the structures and the root mean square distance (RMSD) between the atomic coordinates of these correspondences. Whereas finding the optimal superimposition is a relatively simple task if the set of correspondences is already known [2], finding the optimal superimposition and correspondences simultaneously is NP-hard [3]. Nevertheless, various heuristics have been developed and successfully applied to the pairwise alignment problem [4], [5], [6], [7], [8], [9], [10], [11], [12].

Recently, there has been an increasing focus on the more complex, **m***ultiple* **st**ructure **a**lignment problem (MSTA). Structural alignment of a set of related proteins helps find the conserved cores shared by all or a subset of proteins and gives better insight into the significance of these structural cores than the pairwise alignment. Unfortunately, MSTA is computationally a very difficult problem. Even for a fixed transformation, finding the optimal correspondences among residues from k proteins of average length L takes $O(L^k)$ time under most standard distance measures.

In order to reduce the computational complexity, most approaches build a multiple alignment based on progressively aligning inputs in a pairwise manner [13], [14]. For example, the *center-star* approach used by Gerstein and Levitt [15] maintains a consensus

- *H. Sun, H. Ferhatosmanoglu, and Y. Wang are with the Dept. of Computer Science and Engineering, The Ohio State University, Columbus, OH, USA.*
- *H. Ferhatosmanoglu is also with the Department of Computer Engineering, Bilkent University, Turkey.*
- *A. Sacan is with the School of Biomedical Engineering, Drexel University, Philadelphia, PA, USA.*

This research is partially supported by US National Science Foundation (Grants IIS-0546713 and DBI-0750891)

template, and at each step, a new input structure is aligned to this consensus by pairwise alignment method. Alternatively, one can also construct a consensus template hierarchically using a binary similarity tree, where each leaf represents an input structure, and each internal node aligns the two structures from its children [13], [16]. One of the main limitations of these greedy methods is that following locally (pairwise) optimal solutions may not lead to a globally optimal solution. As a result, these methods are not effective at detecting low levels of similarities, as an incorrect decision committed early on may cause to miss the few correspondences that would have otherwise led to the globally optimal solution.

In contrast to progressive pairwise methods, aligned fragment pair (AFP) chaining methods break each input structure into a set of small *motifs*, such as short fragments of protein backbones [17] or the secondary structure elements (SSEs) [18]. Motifs shared by all proteins are then assembled in a geometrically consistent manner. Since the motifs are much smaller than the whole protein, one can afford to use more accurate methods to align them. Furthermore, using the alignments between motifs as seeds to align the entire structures helps detect partial local similarities among the input structures, yielding *flexible* alignments.

While the AFP methods tend to be more effective at aligning proteins with diverse structures, they still present limitations and challenges. We observe that the performance of the AFP methods rely heavily on the quality of the representation provided by the fragments. Using backbone fragments [17] tend to produce too many motifs and each motif is only constructed by local sequence fragments which hardly reflect spatial similarity; while using SSEs (or relations between SSEs) [18] may miss motifs that are not based on secondary structures. Specifically, we wish to find a *concise* (so that the computational cost remains low), yet *complete* (so that we do not miss important structural similarities) set of motifs. Furthermore, the extension of the seed fragment alignments to global alignments also remain a challenging problem. Currently, the filtering employed on the possible seeds and the geometric constraints imposed during the extension stage, in most cases, speed up the process at the cost of missing better global alignments.

In this paper, we propose and develop a robust MSTA algorithm that addresses the aforementioned limitations and challenges. In particular, for each input protein, we construct a small set of structurally related motifs based on interacting windows in its contact map. The contact map motifs are able to capture features from both SSEs and the residues that do not form distinct SSEs. Additionally, they are spatially constructed to encode geometrical and functional information not available in sequence fragment based motifs. We then develop a novel multi-level extension algorithm that rapidly extends seed alignments from contact-map motifs to global alignments among multiple structures. Finally, we iteratively improve the resulting alignments by an enhanced partial order curve comparison method [19], which further optimizes the correspondences among proteins.

This strategy induces a sensitive and robust automated algorithm that can detect similarities among multiple protein structures even under low similarity conditions. The success of our method is demonstrated on several protein structure datasets that have previously been used under the context of MSTA and that span various structural folds and represent different protein similarity levels. For all of the datasets, our method yields better alignment results compared to other popular MSTA methods in general. Our resulting software is available both as a downloadable binary and as a web service at http://bio.cse.ohio-state.edu/Smolign

2 METHODS

THE objective of our algorithm is to find the largest multiple alignment among k protein structures while maintaining a cumulative error below a threshold ϵ. This error is quantified as the multiple RMSD ($mRMSD$) measure [17] which computes the average of the RMSD values between the aligned residues of a pivot protein p and the corresponding residues of the other proteins:

$$mRMSD_p = \frac{1}{k-1} \sum_{i=1, i \neq p}^{k} RMSD(P_p, P_i) \qquad (1)$$

where P_p denotes the pivot protein and P_i represents each of the k proteins. Variations of this error measure exist, such as using all-pairs average RMSD instead of the average RMSD to a pivot structure, or weighting the contribution of individual residues or individual structures in the calculation of the error measure [20]. For brevity, we have focused our discussion to the $mRMSD$ measure defined above, which is a widely accepted and reported error measure.

A high level description of our algorithm is shown in Figure 1. From a dataset of k protein structures, we first extract contact window patterns from the distance map of each protein. These patterns provide a transformation-invariant representation of local structures. We observe that pairs of contact windows present a good balance between sensitivity and specificity of fragments to be utilized in multiple structure alignment. Therefore, the contact window patterns in a distance map that are in close proximity are paired up into linked motifs, which make up the *Spatial Motifs Library* (SML). Compatible motifs common to all proteins are identified from the SML using a dynamic filtering procedure. An efficient distance-map based alignment method is used to build local seed alignments as a set of correspondences. The local seed

alignments that induce similar 3D transformations and whose combination satisfy a predefined mRMSD threshold are merged to build larger extended seed alignments. To obtain a rigid structure alignment, a single extended seed is refined using the EPO method, an enhanced partial order curve comparison algorithm [19]. To obtain a flexible structure alignment, multiple extended seed alignments that cover different portions of the protein structures are used in the refinement step. In the following sections, we describe each of these steps in detail.

2.1 Construction of the SML

The residue-contact patterns of protein structures are the most conserved features of distantly-related proteins [21], which motivates us to capture and use such patterns for aligning multiple structures. We represent each protein structure using the *distance matrix* [22] of its alpha-carbon atoms. Distance matrix captures the structural and connectivity information and provides a complete representation of the protein structure that is invariant under rigid transformations [23].

The entries of the distance matrix that are less than a predefined threshold (typically 6Å) are denoted as *contact cells* and they correspond to the residues that are in close proximity in the 3D structure. The collection of these cells give the *contact map* of the protein (Figure 1b), which can be used to identify SSE or other structural patterns. Specifically, the fragments along the diagonal are alpha-helices (α), the fragments parallel or perpendicular to the diagonal are parallel and anti-parallel beta-sheets (β^+ and β^-), and other, less regular fragments of residue contacts correspond to small loops (L) and free shapes (F). We utilize the distance and contact maps to extract and classify similar structural motifs that constitute the Spatial Motif Library (SML).

Contact windows. An initial 4×4 sliding window is used to scan the distance map for detecting any of the SSEs and other significant patterns. We then expand the initial size of the captured window row and column-wise simultaneously until such an expansion no longer incorporates a new contact cell.

Note that individual contact windows by themselves do not in general provide a sensitive representation to be used for structural alignment. Because of the regularities in SSEs, many of the contact windows from multiple proteins would align well, but would not necessarily induce a good alignment for the rest of the protein. On the other hand, using pairs of contact windows as seed motifs greatly increases the discrimination power of such motifs. One can use even higher order motifs by combining multiple contact windows; however, this risks being too restrictive and it may not be possible to find such higher order motifs shared by all proteins. Therefore, we use pairs of contact windows as our primary *spatial motifs*, to serve as seed alignments.

Using pairs of structural fragments have previously been utilized by one of the earlier MSTA methods [18], where SSEs are represented as line segments and pairs of SSEs are used to provide seed alignments. Using contact windows instead of SSEs provides a more descriptive representation of motifs and captures spatial arrangements that do not form distinct SSEs.

Spatial Motifs. Pairs of interacting and compatible contact windows are linked to form the *Spatial Motifs* (Figure 1c). A *regular* spatial motif is formed by linking two α helices ($\alpha\alpha$), or an α helix and a β sheet ($\alpha\beta$), or two β sheets ($\beta\beta$). In order to impose that the linked contact windows are interacting in the 3D structure, we further require that the fragments represented by the contact windows are closer than a predefined threshold (typically 13Å), and in the case of β sheets, that they share one of their strands.

Note that for some sets of proteins, the regular motifs formed by α and β contact windows may not be sufficient to induce a global alignment. Moreover, the SSE assignments are error-prone and may not be consistent across the related proteins. In order to handle such cases, we store the *irregular* contact windows from loops (L) and free shapes (F) as part of the SML, and resort to these motifs if the regular motifs do not provide satisfactory alignment seeds.

2.2 Obtaining seed alignments

Alignment of similar motifs from the SML would provide seed alignments around which the rest of the protein structure can be aligned. However, determination of similarity involves the expensive operations of finding residue correspondences and performing structural alignment. We develop several pruning strategies to reduce the number of spatial motifs to be compared. In order to facilitate efficient identification and fast alignment of compatible motifs, we associate each motif with the following features:

- Number of amino acid residues (δ) separating the contact windows along the backbone.
- The minimum Euclidean distance (D) between the amino acid residues of the pairs of contact windows.
- The angle (θ) between the backbone segments in each applied contact window.

Our pruning strategy relies on heuristics using the SSE types, and the D, δ and θ feature values of the motifs. We only perform alignment of motifs that are similar within the thresholds for these features. The thresholds are adjusted dynamically starting from strict similarity and gradually relaxing the threshold values until a desired number of high quality seed alignments are obtained. After the pruning step, we obtain a set of *candidate seeds*, where each seed consists of k similar motifs, with exactly one from each protein.

Alignment of candidate seeds. In the alignment stage, we consider each candidate seed separately and

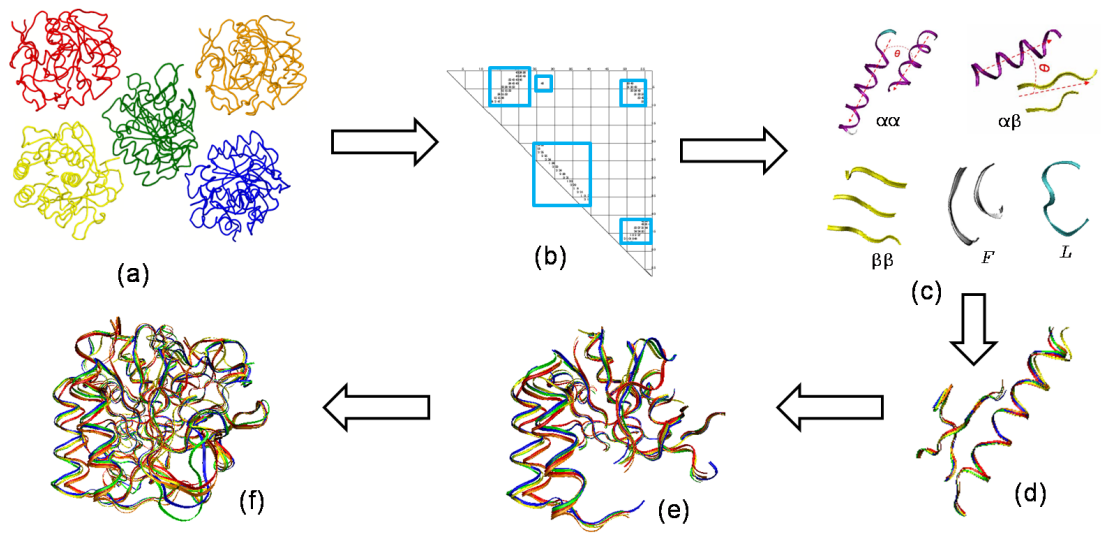


Fig. 1: Overview of the algorithm. (a) Input protein structures. (b) An example contact map. The contact cells are shown as dots in the corresponding matrix entries. The sub-windows are extracted to cover the spatial patterns in the contact map. (c) Spatial Motif Library composed of motifs extracted from the contact maps. (d) Seed alignment of an $\alpha\beta$ motif. (e) Extended seed alignment from compatible seeds. (f) Refined alignment using EPO on the extended seed.

perform alignment of its member motifs to generate and identify the *seed alignments* satisfying the mRMSD criteria. The alignment of the spatial motifs involves identifying residue correspondences and from these correspondences, calculating the superimposition that minimizes the mRMSD measure.

The beta-sheets possess relatively well-defined shapes. Thus, for the $\beta\beta$ category, we simply select the smallest motif to be the *central* motif and slide it over the rest of the motifs in the candidate seed to generate gapless alignments. We then apply Quaternion transformation and rotation [24] based on the correspondences induced by each alignment and identify the seed alignments that satisfy the mRMSD criteria.

For the rest of the motif categories, we utilize the contact windows of the motifs to assign the residue correspondences. The contact window (CW) of a motif is part of the contact map that covers only the residues forming the motif. The alignment of two contact windows (CW_1 and CW_2) is found using the $MaximumOverlap$ algorithm below. The contact windows are slided over each other and each sliding window defines a gapless alignment between the two motifs. The algorithm returns the sliding window that maximizes the number of contacts common to both contact windows as induced by the alignment.

We consider each motif in a candidate seed as the *central* motif and calculate the pairwise alignments with each of the rest of the motifs in the candidate seed. If a contact cell from the central motif's contact window overlaps with a contact cell from every other motif, we note that there is a common correspondence involving a pair of amino acids from each protein.

Algorithm 1: $MaximumOverlap$

Input: contact windows CW_1, CW_2
Output: $bestS$: sliding window with maximum overlap of contacts

```
maxContacts ← 0;
foreach sliding window s aligning CW1 and CW2 do
    count ← 0;
    foreach pair of overlapped cells do
        if both are contact cells then
            count++;
    if count > maxContacts then
        maxContacts ← count;
        bestS ← s;
```

We repeat the alignment procedure, considering each of the motifs as the central motif, and seek the one that gives the maximum number of common correspondences. Based on these correspondences, the Quaternion transformations are calculated to obtain the mRMSD error of the alignment.

Figure 1d shows an example candidate seed from the $\alpha\beta$ category, which includes 5 Serine Protease proteins represented in color. The longest common correspondences of the candidate seed is found to be 34, which gives a seed alignment with an mRMSD of 0.44Å.

2.3 Extending the seed alignments

Each seed alignment contains a small local geometrical motif common to all protein structures and can

be used as a reference to rotate and translate the whole structures. However, we realize that an individual candidate seed may be too small to generate high quality global transformations. Furthermore, some of the seed alignments may induce the same global alignment causing redundant computation. To alleviate these problems, we construct more reliable skeleton structures through merging of compatible seed alignments.

In the ExtendSeed algorithm outlined below, a seed alignment s_i is enriched with the compatible correspondences from other seeds that have similar transformations. A correspondence is added onto s_i so long as it does not conflict with a correspondence already present in s_i and its addition still maintains a structural superposition error below the threshold ($mRMSD < \epsilon$).

Algorithm 2: $ExtendSeed$

Input: S: the set of seed alignments
Input: $s_i \in S$: the seed to be extended
Output: s_i: the extended seed
foreach $s_j \in S$ *and* $s_j \neq s_i$ **do**
  if $\tau_j \approx \tau_i$ **then** //similar transformations
    foreach $cp \in s_j$ **do** //cp: residue correspondence
      if *not Conflicts(cp,*s_i*) and* *mRMSD(*$s_i \cup cp$*)*$< \epsilon$ **then**
        $s_i \leftarrow s_i \cup cp$

Each extended seed combines multiple motifs from the seed alignments and obtains longer high quality correspondences. A larger extended seed provides more reliable basis for the Quaternion transformation and induces a better global alignment with a larger core. In the sample shown in Figure 1e, the seed alignment is extended from 34 (0.44Å) to 134 (1.0Å) common correspondences.

2.4 Refinement by EPO

The extended candidate sets provide correspondences for only certain sections (motifs) of the protein structures, from which pairwise translation and rotation matrices are generated. It still remains to find correspondences for the rest of the structure and optimize the transformations to minimize the global mRMSD. We use the Enhanced Partial Order (EPO) curve comparison algorithm [19] to find common superpositions of the transformed structures and optimize the global rigid-body alignment.

The EPO algorithm has been developed as an improvement over the partial order alignment (POA) methods [25], [26], especially enhancing the sensitivity in detecting low levels of similarity and the ability to handle high dimensional curves. The overall algorithm of EPO is composed of two main stages: the initial construction of a partial order graph (POG) representing the consensus alignment of structures, and a merging stage that refines the POG by merging its nodes while maintaining the constraints defined by the order of residues along each path. Using this update scheme, EPO performs an iterative optimization process, where each iteration generates new correspondences and transformations, which are then used as input to the next iteration. The process is repeated until no improvement in $mRMSD$ is obtained. The details of the EPO algorithm, along with its application to investigation of folding trajectories, are discussed in [19]. Figure 1f shows the final alignment of 5 protein structures; where EPO finds a structural superposition of 243 correspondences with $mRMSD = 1.15$Å.

2.5 Flexible alignments

Introducing flexibility to structural alignment becomes useful for two main reasons. First, a protein may be present in multiple conformational states due to phosphorylation, interaction with other proteins, or ligand binding [27]. Second, distantly related proteins contain twists and bends in their structures that cannot be detected by rigid alignment alone. Because Smolign uses a bottom-up approach starting from local structural motifs, the method introduced thus far can naturally be extended to handle flexibility in alignments. Specifically, we achieve this by building multiple structural cores that cover different areas of the proteins, without restricting that they share the same rigid transformation. The final set of alignments generated in this way not only handle flexibility in the structures, but also can capture sequence order independent alignments.

The $CollectFlexibleSeeds$ algorithm below outlines the process of identifying a complementary set of structural cores from the extended seed alignments produced in Section 2.3. In order to avoid testing an exponential number of different combinations of seeds, we use a heuristic cost measure to focus the grouping of seeds toward combinations that include larger, complementary fragments. For each seed, we quantify the cost of combining it with other seeds by a $mergeCost$, defined as:

$$mergeCost_i = \frac{number\ of\ seeds\ conflicting\ seed_i}{size\ of\ seed_i} \quad (2)$$

We sort the list of seeds by their $mergeCost$ values and starting with the seed that has the smallest $mergeCost$, we combine compatible seeds to cover as much of the proteins as possible. A new seed is combined with the collection of compatible seeds S', only if its inclusion increases the coverage of the correspondence set by a $minFragment$ threshold ($minFragment = 4$ is used as the default value). This ensures that the proteins are not over-fragmented in the final flexible alignment.

Algorithm 3: $CollectFlexibleSeeds$

Input: $S = \{s_i\}$: the set of extended seeds
Output: S': collection of compatible extended seeds

```
Sort S in ascending order of mergeCost;
S' ← {s_0};
for i = 1 ... |S| do
    if mergeCost == 0 then //can be added without conflicts
        S' ← S' ∪ s_i
    else
        s'_i ← s_i \ S' //residues not already covered;
        if |s'_i| ≥ minFragment then
            S' ← S' ∪ s_i
```

After a collection of core alignments is obtained, each core is used to induce an optimized multiple alignment through EPO, as done in Section 2.4. Whenever a residue correspondence conflict arises between the assignments of different cores, the assignment of the larger core is kept. In order to spatially combine the transformations of multiple cores, we take the central protein structure from the first core in the collection as the rigid structure. The transformations of the other cores are calculated in reference to this central structure. The residues that do not have any correspondences are transformed using the transformation of the first core.

3 Experiments

We performed a number of case-based and large scale experiments to demonstrate the capability of Smolign to handle different challenges of MSTA problems. In Section 3.1, we report the results of typical multiple alignment datasets from the literature and discuss how well Smolign handles different spatial data. In Section 3.2, we describe a flexible alignment case in detail. Finally, in Section 3.3, we provide a large scale comparison with other MSTA methods using the Homstrad benchmark [28]. The experiments presented here, along with alignments from the BAliBASE [29] benchmark dataset, are made available on the supplementary website.

We compare the multiple alignments generated by Smolign with those generated by other multiple structure alignment method, namely CE-MC [30], Multiprot [17], MAMMOTH-mult [31], POSA [32], and MASS [18]. CE-MC [30] uses the CE [7] algorithm to perform all-pairwise alignments, which are then progressively combined following the order defined by the UPGMA guide tree [33] of the pairwise alignments. The progressive alignments are refined using Monte Carlo simulations. The CE [7] pairwise alignment algorithm that forms the basis for CE-MC uses short backbone segments as aligned fragment pairs (AFP), which are combined using combinatorial extension.

Multiprot [17] is also a fragment-based multiple structure alignment method. In contrast to the guide-tree approach of CE-MC, it follows a center-star [15] method where each protein is tested as a pivot against which all others are aligned. Multiprot uses a sweeping technique to detect aligned fragments from multiple proteins, enabling Multiprot to detect partial alignments that do not involve all of the input proteins.

MAMMOTH-mult [31] (also referred as MAMMOTH in this report) follows an approach similar to CE-MC [30]. It generates a guide tree from all pairwise alignments, where each pairwise alignment is produced using the MAMMOTH [9] pairwise alignment method. MAMMOTH-mult additionally employs a SIMPLEX [34] optimization of the multiple alignment at each step, to counteract the greediness of the progressive alignment. Like CE-MC and Multiprot, MAMMOTH is a fragment-based alignment method. MAMMOTH uses unit-vector root mean square (URMS) distance [35] between heptapeptide segments as the main mechanism to detect corresponding residues. A method similar to MaxSub [36] is used to find the largest subset of residues that align within a predefined distance threshold (4Å).

The POSA [32] multiple structure alignment program extends the formalism introduced by the FATCAT [37] pairwise structure alignment method. Similar to other structure alignment methods, it starts with identifying a list of aligned fragment pairs (AFP), where each fragment is 8 residues long and the RMSD between the AFPs is defined to be less than a distance threshold (3Å). The structure alignment of these AFPs is represented using a Partial Order Graph, which is a Directed Acyclic Graph. POSA follows a progressive alignment using a guide-tree, similar to CE-MC and Multiprot, but uses single linkage clustering instead of average linkage. POSA has the unique feature of being one of the few multiple structure alignment methods that can generate a flexible alignment.

The MASS [18] multiple structure alignment differs from the other multiple alignment methods in that it considers all the given structures simultaneously, rather than progressive alignment following a guide-tree. MASS uses secondary structure elements as the basic representation of the proteins, and identifies matching SSEs from multiple proteins using Geometric Hashing [38]. Each SSE is represented as a least squares line from its C_α atoms, and each pair of SSEs is represented as two line segments, and the midpoint-distance and angle between them. The type of SSE is also utilized to focus the matching on the most similar SSE segments. Like Multiprot and POSA, MASS is able to detect alignments involving only a subset of the proteins.

Smolign differs from these multiple structure align-

Data set	Members	Average Size	PDB Codes
Set 1 Serine Proteases	5	277	1cseE 1sbnE 1pekE 3prkE 3tecE
Set 2 Calmodulin-like	3	161	1jfjA 1ncx 2sas
Set 3 Tim-barrels	7	391	1btc 1pii 1tml 4enl 5rubA 6xia 7timA
Set 4 2 Helix-Bundle	10	140	1flx 1aep 1bbhA 1bgeB 1le2 1rcb 256bA 2ccyA 2hmzA 3inkC
Set 5 OB fold	15	176	1afp 1b9nA3 1ckmA2 1esfA1 1fr3A 1jic 1tiiD 2tmp 1b7yB2 1bovA 1eif02 1fjgQ 1htp 1sro 2sns

TABLE 1: Protein data sets used for comparing structural alignment methods. *Average Size* is the average number of residues in the proteins in each data set.

ment methods mainly in its use of contact windows as the main representation of proteins. Smolign uses contact windows, which is less restrictive than backbone segments of predefined lengths or backbone segments that form well-defined SSE elements. The filtering employed in Smolign is similar to MASS, except that using contact windows allows additional opportunities for filtering as described in Algorithm 1 above, before a more costly structure superposition is to be employed. Like MASS, Smolign considers all of the protein structures at once, and avoids the local optima caused by the guide-tree based approaches. The refinement step used in Smolign is comparable in its nature to the Partial Order Graph (POG) search used in POSA; Smolign employs the EPO algorithm [19] to refine and extend a multiple alignment of all of the proteins, whereas POSA employs POG search at each of its pairwise iterations. Like POSA, Smolign is able to generate flexible structure alignments.

Using contact windows instead of backbone segments of predefined lengths or segments that form well-defined SSE elements avoids missing structural cores that do not obey these assumptions.

3.1 Sample Alignments

5 protein structural datasets are used to benchmark the performance of our algorithm (See Table 1). These datasets represent different structural folds, span different structural similarity levels, and have previously been used in analysis of multiple structure alignment algorithms. The multiple alignment results for all 5 datasets are compared with those of other popular MSTA methods. In particular, we compare with CE-MC [30], Multiprot [17], MAMMOTH-mult [31], POSA [32], and MASS [18].

We obtained the multiple alignments for each dataset using the online web service provided for these methods. Two vital norms are used for comparing the results: *NCORE*, which is the length of the multiple alignment calculated as the number of amino-acid correspondences, and *mRMSD*, which is an indicator of the alignment quality.

The results for all methods are summarized in Table 2. The POSA algorithm provides two sets of results: flexible and non-flexible alignments. We use the non-flexible alignments for comparison here and use the flexible case in the next sub-section. For the results from MAMMOTH, we count the number of "strict cores" as *NCORE* since "loose cores" reported by MAMMOTH only align partial structures closely. Multiprot allows adjustment of its parameters and returns the most competitive results; we have adjusted its parameters to obtain an accuracy level that matches that of Smolign, in order to make the *NCORE* comparison more meaningful. Specifically, the accuracy values of 3.8Å, 4.4Å, 3.5Å, 3.1Å, and 3.0Å was used for the Multiprot server for datasets 1-5, respectively.

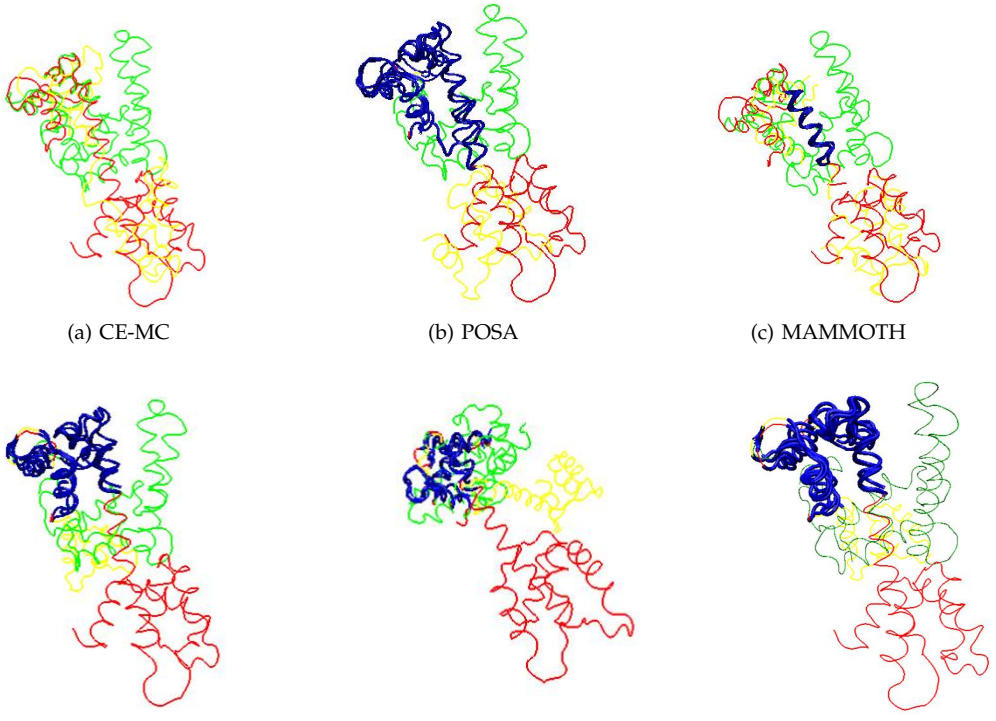
Note that the main objective of our method is to obtain the longest alignment that satisfies a user-defined structural similarity threshold. In some cases, smaller but more conserved alignments may also be biologically important and of interest to the user. Therefore, in the available implementation we provide the top n final alignments, in decreasing order of the alignment lengths. For comparison with other methods, we report here only the top scoring alignment for each dataset in Table 2. The complete set of alignments obtained by Smolign can be viewed and downloaded from the supplementary website.

The 5 proteins in Set 1 belong to the Subtilases family of subtilisin-like serine proteases, that have a common evolutionary origin and share highly similar structures and functional features [39]. All of the compared methods align these proteins reasonably well. Our method provides better alignments than CE-MC, POSA, and Multiprot. POSA has the maximum *NCORE* but incurs a large *mRMSD* cost. MAMMOTH and MASS generate more conservative alignments, that align tightly but have smaller coverage. If the ϵ error threshold in Smolign is reduced from 3Å to 2Å in order to seek more conservative alignments, it is possible to obtain an alignment with *NCORE*=230 and *mRMSD*=0.89Å, which is a longer alignment than that of MAMMOTH, with only a slightly worse *mRMSD*.

Set 2 has only 3 proteins (PDB: 1cnx, 1jfjA, and 2sas), but the aligned motifs are very diverse. CATH [40] classifies 1ncx and 2sas to have one alpha helical domain and 1jfjA to have two alpha helical

Data Set	CE-MC		POSA		MAMMOTH		Multiprot		MASS		Smolign	
	N_{core}	mRMSD	N_{core}	mRMSD	N_{core}	mRMSD	N_{core}	mRMSD	N_{core}	mRMSD	N_{core}	mRMSD
Set 1	244	1.83Å	252	2.08Å	223	0.86Å	237	1.29Å	228	0.97Å	245	1.14Å
Set 2	62	5.80Å	67	2.92Å	15	1.64Å	58	1.92Å	50	1.4Å	59	1.95Å
Set 3	-	-	-	-	-	-	27	2.08Å	30	2.00Å	41	2.08Å
Set 4	-	-	-	-	-	-	22	1.80Å	15	1.80Å	34	1.78Å
Set 5	-	-	-	-	-	-	9	1.27Å	-	-	13	1.74Å

TABLE 2: Comparison of multiple structure alignment methods on sample alignment datasets. In order to obtain comparable results with other methods, a similarity threshold of $\epsilon = 3$Å was used in Smolign. "-" indicates that the respective server did not return any results.



(a) CE-MC (b) POSA (c) MAMMOTH

(d) Multiprot (e) MASS (f) Smolign

Fig. 2: Multiple structure alignments of Set 2 Calmoduline-like proteins by different methods. Each protein is shown in a different color: 1jfj, yellow; 1ncx, red; and 2sas, green. The thick blue portions of the backbones indicate the aligned residues. CE-MC alignment provides the superposed structures, but not the residue correspondences.

domains. The alignments produced by each method is shown in Figure 2. CE-MC and POSA return alignments with inferior *mRMSD* scores, without significant improvement in coverage over other methods. Our method, Multiprot, and MASS align the same domain regions, where our alignment is comparable in both norms to Multiprot. MASS gives a smaller core and a better *mRMSD*. MAMMOTH, as in Set 1, finds a very small conservative core with a worse *mRMSD* than MASS. We are again able to control the accuracy of our results by seeking more conservative alignments that satisfy a smaller mRMSD threshold and obtain an alignment with $NCORE = 48$ and $mRMSD = 1.4$Å when $\epsilon = 1.7$Å, which is comparable to the output of MASS. The Smolign alignment is shown in Figure 2f. The differences in the alignment of this dataset is mainly due to the fact that the progressive pairwise alignment procedure prevents the methods to find the best alignment. While the proteins 1ncx and 2sas are most similar at the EF-hand calcium binding domain (cd00051 in the Conserved Domain Database [41]), 1jfjA and 2sas are most similar at the long alpha-helical segment that connects the two EF-hand domains. An initial alignment of 1jfjA and 2sas, having better global similarity than the other two pairwise alignments, prevents the EF-hand domains of all three proteins to be aligned properly. The center-star alignment procedure used in Multiprot, and the non-progressive alignment methodology of MASS and Smolign avoid this pitfall and give better results. MASS and Smolign capture the common EF-hand domain by using the alignment seeds from the

EF-hand region, and considering all of the proteins simultaneously, extend these seeds to obtain the final alignment core.

Set 3, the Tim-barrels proteins, contains 7 complex structures. Each structure has multiple alpha-helices and beta strands, creating a large number of potential alignment combinations. CE-MC, POSA, and MAMMOTH fail to produce an alignment. Our algorithm not only outperforms both Multiprot and MASS, but also produces an alignment with better spatial continuity. Figure 3 shows that Multiprot aligns less number of structural fragments, whereas MASS produces an over-fragmented alignment core, and only Smolign captures the most complete set of structural fragments, including 3 alpha-helical segments and 4 beta strands. Note that, the Tim-barrel proteins usually contain their enzymatic active sites on the loop regions, frequently on the C-terminal end of the sheets. While it is desirable to detect such functional residues, they are not part of the conserved structural core of the proteins and are not detected by multiple structure alignment methods. Methods based on residue conservation [42] are more appropriate for such an analysis.

Set 4 contains helix-bundle proteins selected from 6 superfamilies, whose skeleton includes four closely packed alpha-helices. It presents a challenge for MSTA methods because of the large dataset size and its structural divergence. CE-MC, POSA, and MAMMOTH again fail to report an alignment. MASS alignment contains a very short helix pair, whereas Multiprot reports either a single long helix or a shorter helix pair depending on the chosen parameters. Smolign consistently outperforms both methods in both norms: it finds a longer alpha-helix pair and a higher quality alignment. Smolign alignment takes under 8 minutes for this dataset.

Set 5 is a very large data set of OB-fold proteins, serving as a stress test for the multiple alignment programs, and the similarity among proteins is extremely low (7% average sequence identity). It is commonly used as a special case to test the sensitivity of MSTA methods. Only our method and Multiprot survive the strain, giving comparable *NCORE* and *mRMSD* trade-offs. The common fold of the OB(oligonuclueotide/oligosaccharide binding)-fold proteins has a five-stranded beta-barrel, capped by an alpha helix [43]. Multiprot finds an alignment involving only two of these beta-strands. Smolign is able align three of these beta-strands common among the 15 proteins in the dataset, at an execution time of 40 minutes.

3.2 Flexible Alignments

The flexible alignment feature of Smolign is demonstrated here using the data set 2, Calmodulin-like proteins. These proteins are composed of two distinct components separated by a long and flexible alpha helix. Due to bending of this alpha helical segment, it is not possible to simultaneously align the two substructures by a rigid alignment (Figure 2f). The best rigid alignment of Smolign aligns 59 residues from the C-terminal domain with an *mRMSD* of 1.95Å. Using this alignment as the anchor, we aggregate compatible cores as described in Section 2.5 to obtain a flexible alignment shown in Figure 4b.

The flexible alignments produced by POSA and Smolign show comparable coverage and quality metrics, while Smolign achieves a less fragmented alignment (Figures 4a and 4b). The main difference of the flexible alignment results comes from the philosophy of applying flexibility. POSA and other MSTA algorithms tend to bend a sequence of fragments multiple times to gain better core size and mRMSD at the cost of loosing structural integrity between aligned fragments. Smolign, on the other hand, strictly maintains spatial consistency of each aligned core, while optimizing for core size and mRMSD. The POSA flexible alignment in Figure 4a breaks the PDB:1cnx structure at 4 locations and does not preserve the spatial relationship of the fragments. Whereas, the Smolign alignment (Figure 4b) consists of only 2 cores whose spatial arrangement is more faithful to the conformation of the structures being aligned and readily yields the interpretation that a single flexible alpha helical segment is responsible for the structural differences among these proteins.

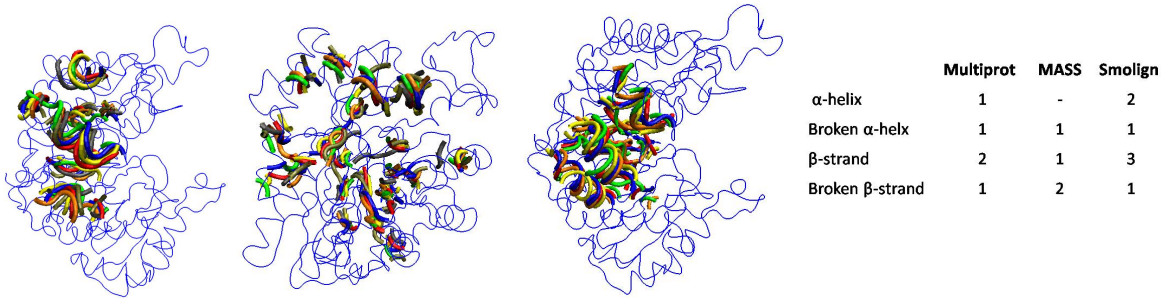
Method	Avg. mRMSD	Avg. Core Size
MATT	2.04	172
Multiprot	1.35	142
MUSTANG	2.67	171
POSA (rigid)	2.00	165
POSA (flexible)	2.22	168
Smolign (rigid)	2.05	174
Smolign (flexible)	2.00	177

TABLE 3: Multiple alignment results for the Homstrad benchmark. mRMSD and core size are averages of all Homstrad datasets. The results (except for those of Smolign) are taken from [45].

3.3 Homstrad Benchmark

Homstrad [28] benchmark dataset contains manually curated pairwise and multiple alignments of highly homologous proteins. The similarity of the aligned proteins is comparable to that of the *family* level in the SCOP [44] hierarchical classification database. Following the experiments by [45] and [32], we use the 399 Homstrad alignments that have more than two structures, to illustrate the performance of Smolign.

The coverage and accuracy of the rigid alignments obtained by Smolign is found comparable to other methods (Table 3). MATT, POSA, and Smolign give similar overall results, with Smolign giving slightly longer alignments comparable or better mRMSD.



	Multiprot	MASS	Smolign
α-helix	1	-	2
Broken α-helx	1	1	1
β-strand	2	1	3
Broken β-strand	1	2	1

(a) Multiprot (b) MASS (c) Smolign (d) Core fragments

Fig. 3: A closer look into the alignment produced by Multiprot, MASS, and Smolign for data set 3, Tim barrels. We only show the complete structure of PDB:4enl as a blue trace. In (d), a helix or strand is considered to be a fragment if its alignment spans more than 5 amino acids and the gaps within the fragment is less than 2.

(a) POSA flexible (b) Smolign flexible

Fig. 4: Rigid and flexible alignments of dataset 2, Calmodulin-like proteins. The rigid/seed core is shown in thick blue trace in each subfigure. (a) Each structure in the rigid alignment is shown in a different color. (b and c) Each alignment core in the flexible alignment is shown in a different color. Blue portion is the alignment core without bending, other colors show alignments after bending. Only 1cnx is shown in full to provide a perspective of the whole structure. The residues of 1jfjA and 2sas that are not part of the alignment are omitted for clarity. Bending occurs on the conjunction points of different colors.

MUSTANG performs worse than others in both mRMSD and core size. Multiprot alignments are more conservative and do not capture the extent of structural fold similarity of the aligned proteins.

While the results for highly similar Homstrad families were consistent among all the methods, Smolign performed comparable to or better than other methods on less similar datasets, such as the *seatoxin* dataset, whose members do not include distinct secondary structure elements, but are composed of many coils and turns. Furthermore, the Smolign flexible alignments are particularly enhanced in detecting multiply concurrent structural motifs while maintaining the spatial continuity of the aligned segments. Comparison of flexible and rigid alignments of the HOMSTRAD datasets identifies 57 cases of flexible alignments. The average coverage of Smolign rigid alignments for these 57 sets were 201 residues (mRMSD=2.19Å). The flexible alignments increase the coverage by 10% (N_{core}=221 residues, mRMSD=2.17Å), with an average of 2.2 bends introduced in each alignment. The rigid and flexible Homstrad alignment results can be accessed on the supplementary web page.

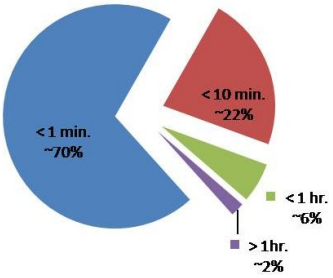


Fig. 5: Running time distribution on 399 Homstrad families. All experiments were performed on an Intel Quad Core 2.66 GHz PC with 4G RAM.

Running time. The execution of Smolign on the Homstrad families takes from seconds to hours, depending on the number, length, and divergence of the structures being aligned and the number of candidate seeds detected for the specified error thresh-

old. Since a rigorous running-time comparison with other methods is not possible due to unavailability of their of software distributions, we summarize the running time of only Smolign in Figure 5. Smolign takes under 1 minute to align 70% of the families and under 10 minutes to align 92% of the families. Of the 8 families that take more than 1 hour to align, 5 families (Homstrad codes: *Cyclodex-gly-tran*, *histone*, *kunitz*, *HLH*, and *RRF*) induce a large number of candidate cores to evaluate; 2 families (*alpha-amylase* and *alpha-amylase-NC*) include a large number of very long peptide chains; and the remaining *rhv* family involves isolated secondary structures which could not be captured in the SML stage and thus forces EPO to execute more iterations to combine the motifs into an optimized rigid alignment.

4 Additional Datasets

We have presented above, the performance of Smolign on a set of commonly used multiple structure alignments and on the Homstrad database. We have also compared the alignments obtained by Smolign against those of some of the popular multiple structure alignment methods. Additional datasets that have been used to benchmark structural alignment methods include SISYPHUS [46], SABmark [47], and BALiBASE [29]. A comprehensive evaluation of the available methods and datasets is beyond the scope of the current study and is left as a future exercise. In this section, we compare Smolign to two of the more recent multiple structure alignment methods, namely MISTRAL [48] and MAPSCI [49].

The MISTRAL structure alignment method [48] uses a piecewise-linear sigmoidal weight function to reward short separations of pairs of amino acids from proteins. A simulated annealing based search over the relative orientations of the proteins is then performed to obtain the translation and rotation matrices that minimize this energy function. MISTRAL follows a center-star multiple alignment approach, by first computing all-pairwise structure alignments and then assigning one of the proteins as the pivot protein to which other proteins are aligned.

The performance of MISTRAL for multiple structure alignments have been demonstrated for four datasets [48]. The first two datasets contain two sets of globins previously considered in [50], and the last two datasets are two groups of proteins from the Homstrad database. The structural alignments generated by Smolign using the default parameters are compared with those reported for MISTRAL are shown in Table 4. MISTRAL has a reported tendency to generate smaller alignments than other methods [48], and this is also observed for datasets 1 and 4, when compared with Smolign. The alignments produced by MISTRAL and Smolign are similar for Set 3, with Smolign giving a slightly longer alignment. Note, however, that Smolign gives a significantly longer alignment with a better mRMSD for Set 2. The residue correspondences reported by MISTRAL are a subset of those reported by Smolign (Figure 6). We attribute the insufficient expansion of the MISTRAL alignment to its protein-centric pairwise evaluation strategy, compared to the motif-centric all-inclusive evaluation used in Smolign. Additional alpha helices and turns detected by Smolign, and the reduced mRMSD are due to the candidate expansion and alignment optimization stages followed in Smolign.

Data Set	Mistral		Smolign	
	N_{core}	mRMSD	N_{core}	mRMSD
Set 1	136	1.4Å	140	1.51Å
Set 2	72	2.1Å	99	1.89Å
Set 3	100	0.7Å	103	0.71Å
Set 4	54	2.0Å	69	2.84Å

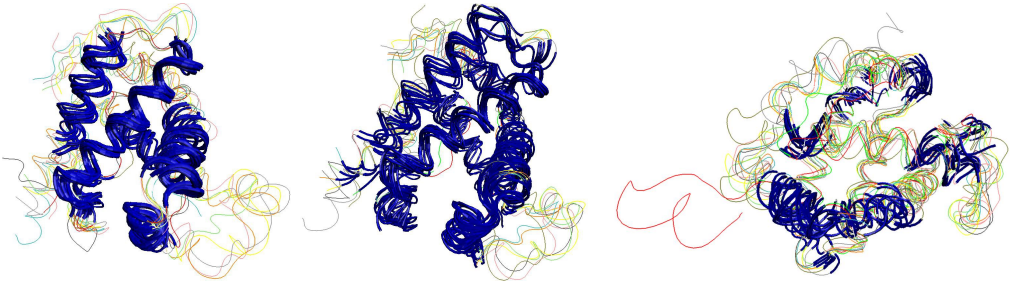
TABLE 4: Comparison of multiple structure alignments obtained by MISTRAL and Smolign on four datasets considered in [48].

MAPSCI [49] is another recent method employing a center-star approach to construct the multiple alignment. The method is quite similar to that described in [51], with the main difference being that MAPSCI works on the C_α coordinates directly, whereas [51] translates the backbone vectors to the origin. Both of these methods work on a consensus pseudo-structure as the average of the proteins being aligned. The sum of the pairwise distances between this consensus structure and each protein in the set is then iteratively minimized to obtain the final alignment.

MAPSCI is reported to produce alignments that compare favorably with the alignments produced by MAMMOTH [9] and MATT [45]. The measurement of the core RMSD is different in MAPSCI than the $mRMSD$ measure reported here, making a direct comparison of the alignment quality difficult. On the other hand, Smolign generally produces alignments with greater coverage than MAPSCI. On a set of 232 HOMSTRAD families considered in [49], MAPSCI produces alignments with an average coverage of 71% (expressed in percent of the length of the shortest protein in each HOMSTRAD family), whereas Smolign produces alignments with an average coverage of 85%.

5 Discussion

We have presented Smolign as a novel multiple protein structure alignment method based on a spatial motif library (SML) generated from residue distance matrices. Smolign provides alignment-order independent results and can generate flexible as well as rigid structural alignments. The alignments produced are comparable to or better than those of other methods, both in alignment quality and coverage.



(a) MISTRAL (b) Smolign (c) difference

Fig. 6: Multiple alignments produced by (a) MISTRAL and (b) Smolign on the dataset of globins from [48]. Residues that are part of the detected alignment are shown in blue. (c) Residues considered part of the alignment by Smolign but not MISTRAL are highlighted in blue.

In the terminology and formalism introduced in [52], Smolign uses an *element based* structure description, as opposed to a *space based* description such as dividing a structure into a grid. Smolign utilizes several element classes, including the contact windows, residue coordinates, and secondary structure elements. The *clustering* of compatible pairs of *structure elements* is done *by use of transformations*, where the element pairs with similar translation and rotation matrices are merged, similar to the SARF program [53] and to the method introduced in [54].

Smolign differs from previous multiple alignment methods in several major aspects. Most importantly, Smolign utilizes contact windows as the basic representation of proteins, from which 3D structural similarities can be identified. Contact windows have previously been used in pairwise structural alignment, DALI [4] being the most known example, but not in multiple structural alignment problem. The main bottleneck in using contact windows for structural alignment is the computational cost of identifying and extending common structural conformations. The problem of finding similar contact sub-windows, known as the Contact Map Overlap (CMO) [55] can be directly translated to a maximum clique problem [56]. Because this is an NP-complete problem [57], several heuristics have been proposed for the pairwise alignment case [58]. Instead of modeling the problem directly as a maximum clique problem, Smolign exploits the additional information contained in the protein structures, such as secondary structure type, and Euclidean distance and angle between backbone segments, greatly reducing the search space.

Other aspects of the novelty of the Smolign include its dynamic filtering of seed alignments that explore the possible candidates in a best-first search and refinement of the alignments by a powerful partial order curve comparison algorithm [19]. Furthermore, Smolign provides the ability to generate flexible alignments, which is not supported by many of the other available methods.

We attribute the success of Smolign to the concise yet complete representation of the input structures it uses to construct the motif library. Pairs of interacting contact map sub-windows provide a good balance between the sensitivity of the representation and the corresponding search space. Through its dynamic filtering and efficient candidate evaluation and expansion algorithms, Smolign handles large and complex datasets where other methods fail to produce any results.

Unless otherwise noted, the results reported here were obtained using the default parameters. These defaults are available on the job submission web site as advanced options. Even though the default parameters achieve competitive results, we allow the interested users to change these parameters to control the quality vs. coverage and the speed vs. accuracy tradeoffs. Of particular importance is the ϵ error threshold, which sets an upper threshold for the mRMSD of the alignment that can be obtained. A tight ϵ error threshold would generate fewer candidate seeds but discover only highly conserved structural motifs, whereas a relaxed ϵ would discover more divergent motifs, at the computational cost of generating many false candidates that need to be evaluated.

We believe that Smolign provides an import step in the advancement of the multiple protein structural alignment, but we acknowledge that it may not give the best or most appropriate results in every single case. While Smolign can be utilized for large scale automated analysis, the use of different alignment programs that are developed under varying assumptions and that use varying representations of proteins, is likely to enrich any given case study. It must also be noted that the currently available multiple structure alignment programs, including Smolign, are geared toward identifying conserved structural cores of proteins, which is an important task in structure classification, fold recognition, and structure prediction problems. On the other hand, they may not be able to identify conservation of individual residue conformations or functional motifs, such as done by LFMPro [59], gSpan [60] and [61].

Smolign is provided both as a web service for fast and convenient access and as a downloadable binary for the more intensive batch tasks. The sample alignments described here and the alignments for Homstrad and BaliBase benchmark datasets are also provided on the supplementary web site.

REFERENCES

[1] M. Sierk and G. Kleywegt, "Deja vu all over again: Finding and analyzing protein structure similarities," *Structure*, vol. 12, no. 12, pp. 2103–2111, 2004.

[2] W. Kabsch, "A discussion of the solution for the best rotation to relate two sets of vectors," *Acta Crystallogr.*, vol. A34, pp. 827–828, 1978.

[3] R. Lathrop, "The protein threading problem with sequence amino acid interaction preferences is np-complete," *Protein Eng.*, pp. 1059–1068, 1994.

[4] L. Holm and C. Sander, "Protein structure comparison by alignment of distance matrices," *J. Mol. Biol.*, vol. 233, pp. 123–138, September 1993.

[5] L. Holm and C. Sander, "3-D lookup: Fast protein structure searches at 90% reliability," *Proc. Ann. Int. Conf. on Intelligent Systems for Molecular*, pp. 179–187, 1995.

[6] W. Taylor and C. Orengo, "SSAP: sequential structure alignment program for protein structure comparison," *Methods Enzymol*, vol. 266, pp. 617–35, 1996.

[7] I. N. Shindyalov and P. E. Bourne, "Protein structure alignment by incremental combinatorial extension (CE) of optimal path," *Protein Engineering*, vol. 11, no. 9, pp. 739–747, 1998.

[8] J. D. Szustakowski and Z. Weng, "Protein structure alignment using a genetic algorithm," *Proteins: Structure, Function, and Bioinformatics*, vol. 38, no. 4, pp. 428–440, 2000.

[9] A. R. Ortiz, C. E. Strauss, and O. Olmea, "MAMMOTH (matching molecular models obtained from theory): An automated method for model comparison," *Protein Sci*, vol. 11, no. 11, pp. 2606–2621, 2002.

[10] A. I. Jewett, C. C. Huang, and T. E. Ferrin, "Minrms: an efficient algorithm for determining protein structure similarity using root-mean-squared-distance," *Bioinformatics*, vol. 19, no. 5, pp. 625–634, 2003.

[11] T. Can and Y.-F. Wang, "CTSS: A robust and efficient method for protein structure alignment based on local geometrical and biological features," *Proc. IEEE Computer Society Conference on Bioinformatics*, pp. 169–179, 2003.

[12] B. Kolbeck, P. May, T. Schmidt-Goenner, T. Steinke, and E.-W. Knapp, "Connectivity independent protein-structure alignment: a hierarchical approach." *BMC Bioinformatics*, vol. 7, pp. 510–530, 2006.

[13] W. R. Taylor, T. P. Flores, and C. A. Orengo, "Multiple protein structure alignment," *Protein Science*, vol. 3, pp. 1858–1870, 1994.

[14] D. F. Feng and R. F. Doolittle, "Progressive sequence alignment as a prerequisite to correct phylogenetic trees." *J Mol Evol*, vol. 25, no. 4, pp. 351–360, 1987.

[15] M. Gerstein and M. Levitt, "Comprehensive assessment of automatic structural alignment against a manual standard, the scop classification of proteins," *Protein Science*, vol. 7, pp. 445–456, 1998.

[16] R. Russell and G. Barton, "Multiple protein sequence alignment from tertiary structure comparison: assignment of global and residue confidence levels," *Proteins*, vol. 14, no. 2, pp. 309–323, 1992.

[17] M. Shatsky, R. Nussinov, and H. J. Wolfson, "MultiProt – a multiple protein structural alignment algorithm," *WABI '02: Proceedings of the Second International Workshop on Algorithms in Bioinformatics*, pp. 235–250, 2002.

[18] O. Dror, H. Benyamini, R. Nussinov, and H. J. Wolfson, "Multiple structural alignment by secondary structures: Algorithm and applications," *Protein Science*, vol. 12, pp. 1492–2507, 2003.

[19] H. Sun, H. Ferhatosmanoglu, M. Ota, and Y. Wang, "Enhanced partial order curve comparison over multiple protein folding trajectories." *Comput Syst Bioinformatics Conf.*, pp. 229–310, 2007.

[20] X. Wang and J. Snoeyink, "Multiple structure alignment by optimal rmsd implies that the average structure is a consensus." *Comput Syst Bioinformatics Conf*, pp. 79–87, 2006.

[21] A. Lesk and C. Chothia, "How different amino acid sequences determine similar protein structures: I. the structure and evolutionary dynamics of the globins," *J. Mol. Biol.*, vol. 136, pp. 225–270, 1980.

[22] J. Richardson, "The anatomy and taxonomy of protein structure," *Adv. Protein Chem.*, vol. 34, pp. 167–339, 1981.

[23] T. Havel, I. Kuntz, and G. Crippen, "The theory and practice of distance geometry," *Bull. Math. Biol.*, vol. 45, p. 665720, 1983.

[24] J. C. Hart, G. K. Francis, and L. H. Kauffman, "Visualizing quaternion rotation," *ACM Trans. Graph.*, vol. 13, no. 3, pp. 256–276, 1994.

[25] C. Lee, C. Grasso, and M. Sharlow, "Multiple sequence alignment using partial order graphs," *Bioinformatics*, vol. 18, no. 3, pp. 452–464, 2002.

[26] C. Grasso and C. Lee, "Combining partial order alignment and progressive multiple sequence alignment increases alignment speed and scalability to very large alignment problems," *Bioinformatics*, vol. 20, no. 10, pp. 1546–1556, June 2004.

[27] C. Lemmen, T. Lengauer, and G. Klebe, "Flexs: A method for fast flexible ligand superposition," *J. Medicinal Chem.*, vol. 41, pp. 4502–4520, 1998.

[28] K. Mizuguchi, C. M. Deane, T. L. Blundell, and J. P. Overington, "HOMSTRAD: A database of protein structure alignments for homologous families," *Protein Sci*, vol. 7, no. 11, pp. 2469–2471, 1998.

[29] P. O. Thompson JD, Plewniak F, "Balibase: a benchmark alignment database for the evaluation of multiple alignment programs," *Bioinformatics*, vol. 15, no. 1, pp. 87–88, 1999.

[30] C. Guda, S. Lu, E. D. Scheeff, P. E. Bourne, and L. N. Shindyalov, "CE-MC: a multiple protein structure alignment server," *Nucleic Acids Research*, vol. 32, pp. W100–W103, 2004.

[31] D. Lupyan, A. Leo-Macias, and A. R. R. Ortiz, "A new progressive-iterative algorithm for multiple structure alignment." *Bioinformatics*, pp. 3255–3263, June 2005.

[32] Y. Ye and A. Godzik, "Multiple flexible structure alignment using partial order graphs," *Bioinformatics*, vol. 21, no. 10, pp. 2362–2369, 2005.

[33] P. H. Sneath and R. R. Sokal, "Numerical taxonomy," *Nature*, vol. 193, pp. 855–860, Mar 1962.

[34] G. J. Barton and M. J. Sternberg, "A strategy for the rapid multiple alignment of protein sequences. confidence levels from tertiary structure comparisons." *J Mol Biol*, vol. 198, no. 2, pp. 327–337, Nov 1987.

[35] K. Kedem, L. Chew, and R. Elber, "Unit-Vector RMS(URMS) as a Tool to Analyze Molecular Dynamics Trajectories," *Proteins: Structure, Function and Genetics*, vol. 37, pp. 554–564, 1999.

[36] N. Siew, A. Elofsson, L. Rychlewski, and D. Fischer, "Maxsub: an automated measure for the assessment of protein structure prediction quality." *Bioinformatics*, vol. 16, no. 9, pp. 776–785, Sep 2000.

[37] Y. Ye and A. Godzik, "Flexible structure alignment by chaining aligned fragment pairs allowing twists," *Bioinformatics*, vol. 19, pp. ii246–ii255, 2003.

[38] R. Nussinov and H. J. Wolfson, "Efficient detection of three-dimensional structural motifs in biological macromolecules by computer vision techniques." *Proc Natl Acad Sci U S A*, vol. 88, no. 23, pp. 10 495–10 499, Dec 1991.

[39] R. J. Siezen and J. A. Leunissen, "Subtilases: the superfamily of subtilisin-like serine proteases." *Protein Sci*, vol. 6, no. 3, pp. 501–523, Mar 1997. [Online]. Available: http://dx.doi.org/10.1002/pro.5560060301

[40] C. A. Orengo, A. D. Michie, S. Jones, D. T. Jones, M. B. Swindells, and J. M. Thornton, "CATH–A hierarchic classification of protein domain structures," *Structure*, vol. 5, no. 8, pp. 1093 –1108, 1997.

[41] A. Marchler-Bauer, S. Lu, J. B. Anderson, F. Chitsaz, M. K. Derbyshire, C. DeWeese-Scott, J. H. Fong, L. Y. Geer, R. C. Geer, N. R. Gonzales, M. Gwadz, D. I. Hurwitz, J. D. Jackson, Z. Ke, C. J. Lanczycki, F. Lu, G. H. Marchler, M. Mullokandov, M. V. Omelchenko, C. L. Robertson, J. S. Song, N. Thanki, R. A. Yamashita, D. Zhang, N. Zhang, C. Zheng, and S. H. Bryant, "Cdd: a conserved domain database for the functional annotation of proteins." *Nucleic Acids Res*, vol. 39,

no. Database issue, pp. D225–D229, Jan 2011. [Online]. Available: http://dx.doi.org/10.1093/nar/gkq1189
[42] A. Armon, D. Graur, and N. Ben-Tal, "Consurf: an algorithmic tool for the identification of functional regions in proteins by surface mapping of phylogenetic information." *J Mol Biol*, vol. 307, no. 1, pp. 447–463, Mar 2001. [Online]. Available: http://dx.doi.org/10.1006/jmbi.2000.4474
[43] A. G. Murzin, "Ob(oligonucleotide/oligosaccharide binding)-fold: common structural and functional solution for non-homologous sequences." *EMBO J*, vol. 12, no. 3, pp. 861–867, Mar 1993.
[44] A. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia, "SCOP: A structural classification of proteins database for the investigation of sequences and structures," *J. Mol. Biol.*, vol. 247, pp. 536–540, 1995.
[45] M. Menke, B. Berger, and L. Cowen, "Matt: Local flexibility aids protein multiple structure alignment," *PLOS Computational Biology*, vol. 4, no. 1, p. e10, 2008.
[46] A. Andreeva, A. Prli, T. J. P. Hubbard, and A. G. Murzin, "Sisyphus–structural alignments for proteins with non-trivial relationships." *Nucleic Acids Res*, vol. 35, no. Database issue, pp. D253–D259, Jan 2007. [Online]. Available: http://dx.doi.org/10.1093/nar/gkl746
[47] I. V. Walle, I. Lasters, and L. Wyns, "Sabmark–a benchmark for sequence alignment that covers the entire known fold space." *Bioinformatics*, vol. 21, no. 7, pp. 1267–1268, Apr 2005. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/bth493
[48] C. Micheletti and H. Orland, "Mistral: a tool for energy-based multiple structural alignment of proteins." *Bioinformatics*, vol. 25, no. 20, pp. 2663–2669, Oct 2009. [Online]. Available: http://dx.doi.org/10.1093/bioinformatics/btp506
[49] I. Ilinkin, J. Ye, and R. Janardan, "Multiple structure alignment and consensus identification for proteins." *BMC Bioinformatics*, vol. 11, p. 71, 2010. [Online]. Available: http://dx.doi.org/10.1186/1471-2105-11-71
[50] A. S. Konagurthu, J. C. Whisstock, P. J. Stuckey, and A. M. Lesk, "Mustang: A multiple structural alignment algorithm," *Proteins: Structure, Function, and Bioinformatics*, vol. 64, no. 3, pp. 559–574, 2006. [Online]. Available: http://dx.doi.org/10.1002/prot.20921
[51] J. Ye and R. Janardan, "Approximate multiple protein structure alignment using the sum-of-pairs distance." *J Comput Biol*, vol. 11, no. 5, pp. 986–1000, 2004.
[52] I. Eidhammer, I. Jonassen, and W. R. Taylor, "Structure comparison and structure patterns." *J Comput Biol*, vol. 7, no. 5, pp. 685–716, 2000. [Online]. Available: http://dx.doi.org/10.1089/106652701446152
[53] N. N. Alexandrov, K. Takahashi, and N. Go, "Common spatial arrangements of backbone fragments in homologous and non-homologous proteins." *J Mol Biol*, vol. 225, no. 1, pp. 5–9, May 1992.
[54] L. P. Chew, D. Huttenlocher, K. Kedem, and J. Kleinberg, "Fast detection of common geometric substructure in proteins." *J Comput Biol*, vol. 6, no. 3-4, pp. 313–325, 1999. [Online]. Available: http://dx.doi.org/10.1089/106652799318292
[55] A. Godzik, J. Skolnick, and A. Kolinski, "Regularities in interaction patterns of globular proteins." *Protein Eng*, vol. 6, no. 8, pp. 801–810, Nov 1993.
[56] D. Strickland, E. Barnes, and J. Sokol, "Optimal protein structure alignment using maximum cliques," *Operations Research*, vol. 53, pp. 389–402, 2005.
[57] D. Goldman, S. Istrail, and C. Papadimitriou, "Algorithmic aspects of protein structure similarity," *In: Proc. 40th Annual IEEE Sympos. Foundations Comput. Sci.. IEEE Computer Society, Los Alamitos*, pp. 512–522, 1999.
[58] W. Pullan, "Protein structure alignment using maximum cliques and local search," *Advances in Artificial Intelligence, LNCS*, vol. 4830, pp. 776–780, 2007.
[59] A. Sacan, O. Ozturk, H. Ferhatosmanoglu, and Y. Wang, "Lfm-pro: A tool for detecting significant local structural sites in proteins," *Bioinformatics*, vol. 23, no. 6, pp. 709–716, 2007.
[60] X. Yan and J. Han, "gSpan: Graph-based substructure pattern mining," in *Proc. 2002 Int. COnf. Data Mining (ICDM'02)*, Maebashi, Japan, Dec 2002, pp. 721–724.
[61] D. Bandyopadhyay, J. Huan, J. Prins, J. Snoeyink, W. Wang, and A. Tropsha, "Identification of family-specific residue packing motifs and their use for structure-based protein function prediction: I. method development." *J Comput Aided Mol Des*, vol. 23, no. 11, pp. 773–784, Nov 2009. [Online]. Available: http://dx.doi.org/10.1007/s10822-009-9273-4

Hong Sun is a PhD candidate in the Department of Computer Science and Engineering at The Ohio state University and currently working as a research scientist at SRA international Inc(NIEHS contractor). His research interests include protein sequence and structure alignment, biomedical data mining, and information retrieval.

Ahmet Sacan received his B.Sc. degrees in Computer Science and in Cellular and Molecular Biology from University of Michigan, Ann Arbor, USA, in 2001; and his Ph.D. in Computer Engineering from the Middle East Technical University, Turkey, in 2008. He is currently an Assistant Professor at Drexel University, School of Biomedical Engineering. His research and teaching interests include structural bioinformatics, microRNA and mRNA expression analysis, biomedical image analysis, object tracking, data mining, database indexing methods, multimedia databases, software engineering for web applications, and distance learning technologies.

Yusu Wang obtained Her M.S and Ph.D degree from Duke Univ., and B.S. degree from Tsinghua Univ. Before joining The Ohio State University, she was a post-doctoral researcher at Geometric Computing lab in Stanford Univ. from 2004-2005. she received DOE (Dept. of Energy) Career award in 2006, and NSF (National Science Foundation) Career award in 2008. She is currently on the editorial board of Journal of Computational Geometry (JoCG). Her research interests including Computational geometry and topology, Shape analysis, Geometric computing, and Computational biology. Her research projects are funded by the NSF and DOE.

Hakan Ferhatosmanoglu received his B.S. degree from Computer Science, Bilkent University, Ankara, Turkey in 1997 and Ph.D. degree from University of California, Santa Barbara in 2001. Currently, he is an Associate Professor in the Department of Computer Science and Engineering at The Ohio State University. His research interests focus on Database Systems and Applications, Biomedical Informatics, High-Performance Data Management, Scientific, Multimedia, and high dimensional databases and Social Networks.